計算機援用セキュリティシステムの構成方法と安全性に関する一考察

神農 泰圭[†1]　兼子 拓弥[†2]　高橋 健太[†3]　尾形 わかは[†4]　西垣 正勝[†5]

概要：CPU の計算機能力は日々向上する．計算機を利用した攻撃に耐性を持たせるためには，秘密情報のエントロピを CPU の進化に伴って増加させる必要がある．ここで，CPU の進化は攻撃者だけではなく，正規ユーザにも恩恵を与える．実際に，正規ユーザの計算機能力を用いてセキュリティ強化を達成したシステムが提案されている．本稿では，それらを計算機援用セキュリティシステム（CASS）と呼ぶ。CASS は，エントロピの観点から「秘密情報補完」型と「秘密情報拡張」型に分類される．本稿では，共通鍵暗号，公開鍵暗号，認証コードに対する秘密情報補完型 CASS の構成法を提案し，前者 2 つの構成に対してはその安全性証明を行う．

キーワード：暗号，計算機援用セキュリティ，共通鍵暗号，公開鍵暗号，安全性証明

A Study on Construction and Security of Computer Aided Security System

Yasuyoshi Jinno[†1]　Takuya Kaneko[†2]　Kenta Takahashi[†3]　Wakaha Ogata[†4]
Masakatsu Nishigaki[†5]

Abstract: CPU performance is making progress day by day. To be resistant to attacks even if CPU performance makes progress, we need to increase entropy of secret information. Progress of CPU performance gives benefit not only to attackers but also legitimate users. In fact, security systems specifically designed so that legitimate users enhance security by using CPU performance are already present. In this paper, we call the systems “Computer Aided Security System” and classify the systems into two types in terms of entropy. We propose a construction and provide the security proof of the systems to some primitives in one type.

Keywords: Cryptography, Computer-Aided Security, Symmetric Key Encryption, Public Key Encryption, Security Proof

1. はじめに

CPU の計算機能力は日々向上する．計算機を利用した攻撃に耐性を持たせるためには，秘密情報のエントロピを CPU の進化に伴って増加させる必要がある．しかし，CPU の計算機能力の進化は攻撃者だけでなく，正規ユーザにも恩恵を与えるものであり，本来，攻撃者のみが有利になるものではない．正規ユーザもその時代の計算機能力を用いて秘密情報の運用能力を高めることができれば，CPU の計算機能力の進化による攻撃者の攻撃能力の増加に対抗する手段となり得る．実際に，正規ユーザの計算機能力を用いて認証や暗号のセキュリティ強化を達成した方式として，計算機援用ユーザ認証[1]，パスワードベース鍵生成関数 bcrypt[2]，PBKDF2[3]が提案されている．著者らは，このような「正規ユーザの計算機能力を用いて秘密情報のエントロピを上乗せする」というアプローチによるセキュリティ確保の概念を計算機援用セキュリティと呼ぶ．計算機援用セキュリティシステムは，エントロピの観点から，「秘密情報補完」型と「秘密情報拡張」型の 2 つのタイプに分類される．本稿では，共通鍵暗号，公開鍵暗号，認証コード（ユーザ認証）のプリミティブに対する秘密情報補完型システムの構成法を提案し，前者 2 つの構成に対してはその安全性証明までを行う．

†1 静岡大学大学院総合科学技術研究科
Graduate School of Integrated Science and Technology, Shizuoka University
†2 静岡大学大学院情報学研究科
Graduate School of Informatics, Shizuoka University
†3 株式会社日立製作所研究開発グループセキュリティ研究部
Hitachi, Ltd, R&D Group, Security Research Dept.
†4 東京工業大学工学院
School of Engineering, Tokyo Institute of Technology
†5 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

2. 計算機援用セキュリティシステム

計算機援用セキュリティシステムの具体的なインスタンスである計算機援用ユーザ認証，パスワードベース鍵生成関数 bcrypt，PBKDF2 について説明し，計算機援用セキュリティシステムの 2 つのタイプを定義する．

2.1 計算機援用ユーザ認証

計算機援用ユーザ認証とは，認証情報の一部が正規ユーザにも未知となっており，正規ユーザ自身が総当たり試行によってその未知情報を求めた上で認証を行う．

認証手順

1. 登録フェーズにて，$H(H(p))$が認証サーバに登録され

ている．

2. サーバはクライアント端末に，$H(H(p))$を送信する．
3. ユーザはクライアント端末にp_uを入力する．
4. クライアント端末は$H(H(p_u|p_r))$と，$H(H(p))$が一致するようなp_rを総当たり試行によって探索する．
5. クライアント端末は$H(p_u|p_r)$をサーバに送信する．
6. サーバは$H(H(p_u|p_r))$と$H(H(p))$が一致したらユーザを認証する．

ここで，p_uはユーザが所持すべき秘密情報，p_rは総当たり試行によって求める未知情報，$p = p_u|p_r$は認証情報である．記号「｜」はビット列の連結，$H(\cdot)$はハッシュ関数である．p_u，p_r，pのビット長をそれぞれk_u，k_r，kとする．

p_uを所持している正規ユーザは，未知情報p_rのみを総当たり試行によって求める形となるため，負担する計算コストは最大で2^{k_r}回である．一方，攻撃者にとってはp_uとp_rの両者が未知であるため，なりすましに必要となる総当たり試行は最大で$2^{k_u+k_r}$回の計算コストとなる．例えば文献[1]では，正規ユーザの計算コストが 1 秒以内，攻撃者の計算コストが 1 年以上となるように，p_uとp_rのエントロピ（ビット長）を決定している．

2.2 パスワードベース鍵生成関数

bcrypt および PBKDF2 は，パスワードを入力として暗号鍵を出力する関数であり，入力（パスワード）から出力（暗号鍵）を求めるにあたっての反復演算回数を可変とすることにより，鍵生成に要する時間をコントロールできるように設計されている．鍵生成に時間を要する分，攻撃者の攻撃試行の速度を減速することができ，これによって暗号鍵の生成源（パスワード）のエントロピ不足を補っている．

bcrypt の暗号鍵生成手順

1. パスワードp_u，ソルトs，反復演算回数cを入力する．
2. パスワードp_u，ソルトsを用いて，あらかじめ定められた初期値を鍵スケジューリング関数によって加工する．
3. パスワードp，ソルトsを用いて，2 の結果を鍵スケジューリング関数によって加工する．
4. 3 を2^c回繰り返す．
5. 4 で得られた結果を blowfish 暗号[4]の鍵として，あらかじめ定められた文字列を ECB モードで 64 回暗号化し，その結果を暗号鍵pとして出力する．

PBKDF2 の暗号鍵生成手順

1. パスワードp_u，ソルトs，反復演算回数cを入力する．
2. p_uを鍵として用いる鍵付きハッシュ関数にsを入力して，U_1を得る．
3. p_uを鍵として用いる鍵付きハッシュ関数にU_1を入力して，U_2を得る．
4. p_uを鍵として用いる鍵付きハッシュ関数にU_2を入力して，U_3を得る．これをU_cを得るまで繰り返す．
5. U_1~U_cの排他的論理和を求め，これを暗号鍵pとして出力する．

p_uを所持している正規ユーザは，2^c回（bcrypt）あるいはc回（PBKDF2）の反復演算によって暗号鍵が生成される．一方，攻撃者にとってはp_uが未知であるため，なりすましに必要となる反復演算は2^{k_u+c}回（bcrypt）あるいは$c \cdot 2^{k_u}$回（PBKDF2）となる．ここで，k_uはp_uのエントロピ（ビット長）である．暗号鍵pのビット長は blowfish 暗号関数（bcrypt）あるいはハッシュ関数（PBKDF2）の出力長である．

2.3 計算機援用セキュリティシステムのタイプ

計算機援用セキュリティシステムは，正規ユーザに計算コストを負担させることによって，k_uビットの入力p_u（ユーザが所持している秘密情報）からkビットの出力p（認証情報や暗号鍵）を生成する．計算機援用セキュリティシステムはエントロピの観点から次の 2 つのタイプに分類することができる．

秘密情報補完型

「kビットの真正乱数pが秘密情報（認証情報や暗号鍵）として設定された後，正規ユーザがその一部を忘却してしまい，k_uビット（$k_u < k$）の秘密情報p_uのみを所持している」という状況における計算機援用セキュリティシステム．正規ユーザは，計算コストを負担することによって，k_uビットの入力p_uからkビットの真正乱数pを復元する．秘密情報のエントロピは，k_uビットからkビットに「実質的に」増加する．計算機援用ユーザ認証がこの典型である．

秘密情報拡張型

「k_uビットの真正乱数p_uが秘密情報（認証情報や暗号鍵）のシードとして設定されており，正規ユーザがそのシードp_uを所持している」という状況における計算機援用セキュリティシステム．正規ユーザは，計算コストを負担することによって，k_uビットの真正乱数p_uをkビットの秘密情報pに伸張する．秘密情報pの真のエントロピ（情報理論的な意味でのエントロピ）はk_uビットのままであるが，「疑似エントロピ（攻撃者の総当たり攻撃にかかる計算コスト）」がkビットに増加する．パスワードベース鍵生成関数がこの典型である．

本稿では，秘密情報補完型の計算機援用セキュリティシステムの構成法とその安全性証明を扱う．

3. 秘密情報補完型計算機援用セキュリティシステムの構成

本章では，共通鍵暗号，公開鍵暗号，認証コード（ユーザ認証）のプリミティブに対する秘密情報補完型計算機援用セキュリティシステムの構成法を説明する．

3.1 計算機援用共通鍵暗号

共通鍵暗号プリミティブに対する秘密情報補完型計算機援用セキュリティシステムの構成法を説明する．以降，本システムを計算機援用共通鍵暗号と呼ぶ．計算機援用共通

鍵暗号は，既知平文攻撃（KPA）に対して安全な任意の共通鍵暗号アルゴリズムを用いて構成される．計算機援用共通鍵暗号においては，平文と暗号文の 1 ペアが「検証情報」として公開される（KPA 安全である共通鍵暗号アルゴリズムを用いているため，検証情報を公開しても安全性は損なわれない）．

3.1.1 準備

共通鍵暗号アルゴリズム

共通鍵暗号は 3 つのアルゴリズム（Gen,Enc,Dec）からなり，Gen は1^k（kはセキュリティパラメータ）を入力とし，秘密鍵skを出力するアルゴリズム，Enc はskと平文mを入力として暗号文cを出力するアルゴリズム，Dec はskとcを入力としてmを出力するアルゴリズムである．

既知平文攻撃

既知平文攻撃者は，共通鍵暗号アルゴリズムに対する攻撃を行うにあたり，ランダムな平文と暗号文のペア$(m_1, c_1), \cdots, (m_q, c_q)$を利用することができる．

3.1.2 計算機援用共通鍵暗号の手順

鍵と検証情報の生成手順

1. Gen を実行し，kビットの秘密鍵skが生成される．
2. Enc を実行し，skを用いて任意の平文mを暗号化して暗号文cを得る．検証情報(m, c)を公開しておく．
3. $sk = sk_u|sk_r$の内，sk_uのみがユーザに渡される．

暗号化または復号の手順

1. ユーザは端末にsk_uを入力する．
2. 端末はsk_1を選び，$sk_u|sk_1$を秘密鍵として，検証情報mを暗号化してc_1を得る．$c_1 = c$ならば，$sk_u|sk_1 = sk$である．$c_1 \neq c$ならば$sk_u|sk_1 \neq sk$であるため，3 へ進む．
3. 端末はsk_2を選び，$sk_u|sk_2$を秘密鍵として，検証情報mを暗号化して，c_2を得る．$c_2 = c$ならば，$sk_u|sk_2 = sk$である．$c_2 \neq c$ならば$sk_u|sk_2 \neq sk$であるため，4 へ進む．
4. 端末は，検証情報を利用して，$sk_u|sk_i = sk$となるまで（$sk_i = sk_r$となるsk_iが見つかるまで）sk_iの総当たりを繰り返し，skを復元する．
5. 端末は，復元されたskを用いて Enc, Dec を実行する．

ここで，記号「｜」はビット列の連結を表す．

3.2 計算機援用公開鍵暗号

公開鍵暗号プリミティブに対する秘密情報補完型計算機援用セキュリティシステムの構成法を説明する．以降，本システムを計算機援用公開鍵暗号と呼ぶ．計算機援用公開鍵暗号は，擬似乱数生成器[5]と，選択平文攻撃（CPA）に対して安全な公開鍵暗号アルゴリズムを用いて構成される．計算機援用公開鍵暗号においては，公開鍵が「検証情報」としても利用される（CPA 安全である公開鍵暗号アルゴリズムを用いているため，検証情報を公開しても安全性は損なわれない）．

3.2.1 準備

擬似乱数生成器

疑似乱数生成器は，多項式時間で計算可能な確定的関数アルゴリズム$PRG: \{0,1\}^k \to \{0,1\}^{k_l(k)}$であり，$k$ビットのシード$s \in \{0,1\}^k$の入力に対し，$k_l(k)$ビットの疑似乱数列を出力する．ここで，$k_l(k)$は$k$の多項式であり，$k_l(k) > k$が成立する．

定義 1

「kビットのシードsをPRGに入力することによって得られる$k_l(k)$ビットの疑似乱数列r」と「$k_l(k)$ビットの真正乱数列r」を識別することを目的とする攻撃者Bを考える．PRGに対するBのアドバンテージを

$$Adv_B^{PRG}(k) = |\Pr[B(r) \to 1 | r \leftarrow \{0,1\}^{k_l(k)}] - \Pr[B(r) \to 1 | r \leftarrow PRG(\{0,1\}^k)]|$$

と定義する．ここで，$B(r) \to 1$は，Bが「r を疑似乱数であると判定した」ことを表している．いかなる確率的多項式時間アルゴリズムBに対しても，$Adv_B^{PRG}(k) < \varepsilon(k)$が成立するとき，$PRG$は計算量的に安全な擬似乱数生成器であるという．

公開鍵暗号アルゴリズム

公開鍵暗号は，4 つのアルゴリズム(Gen1,Gen2,Enc, Dec)からなり，Gen1 は$k_l(k)$ビットの真正乱数（$k_l(k)$はセキュリティパラメータ）を入力とし，秘密鍵skを出力するアルゴリズム，Gen2 はskを入力とし，公開鍵pkを出力するアルゴリズム，Enc はpkと平文mを入力とし，暗号文cを出力するアルゴリズム，Dec はskとcを入力とし，mあるいは$\perp$（復号不可）を出力するアルゴリズムである．

選択平文攻撃

選択平文攻撃者は，公開鍵暗号アルゴリズムに対する攻撃を行うにあたり，公開鍵pkを用いて，任意の平文m_iに対する暗号文c_iを利用することができる．

定義 2

「暗号文を識別することを目的とした選択平文攻撃（IND-CPA）」を例に採り，公開鍵暗号アルゴリズムの安全性を説明する．

攻撃者Aに対する挑戦者 B を設定し，Aと B の間で実行される次のようなゲーム G（IND-CPA ゲーム）を構成する：①Bは，公開鍵pkと秘密鍵skのペアを生成し，②Aにpkを入力する．③Aは 2 つの平文m_0, m_1をBに返す．④Bはランダムにチャレンジビット$b \in \{0,1\}$を選んだ上（0または1のいずれかが等確率で選ばれる）で，$c^* \leftarrow Enc(pk, m_b)$を計算して，⑤$A$にチャレンジ暗号文$c^*$を返す．⑥$A$は，$B$が④で選んだチャレンジビットの値（0か1か）を推測して，その答え$b^* \in \{0,1\}$をBに返答する．Aが 1/2 よりも高い確率でbの値を正答できれば，Aの勝ち．

上記の IND-CPA ゲームに対する攻撃者Aのアドバンテージを

$$Adv_A^{IND-CPA}(k_l(k)) = |\Pr[b = b^*] - \frac{1}{2}|$$

と定義し，いかなる確率的多項式時間アルゴリズムAに対しても，$Adv_A^{IND-CPA}(k_l(k)) < \varepsilon(k_l(k))$が成立するとき，公開鍵暗号アルゴリズムは IND-CPA 安全であるという．

3.2.2 計算機援用公開鍵暗号の手順

鍵と検証情報の生成手順

1. kビットの秘密情報pを擬似乱数生成器のシードとして入力し，$k_l(k)$ビットの疑似乱数列を得る．
2. 1 の出力を Gen1 に入力し，秘密鍵skを得る．
3. skを Gen2 に入力し，公開鍵（かつ検証情報）pkを生成する．pkは公開される．
4. $p = p_u|p_r$の内，p_uのみがユーザに渡される．

暗号化の手順

1. 端末は，公開鍵pkを用いて Enc を実行する．

復号の手順

1. ユーザは端末にp_uを入力する．
2. 端末はp_1を選び，$p_u|p_1$を擬似乱数生成器に入力する．その出力を Gen1 に入力してsk_1を得る．更に Gen2 を用い，sk_1よりpk_1を生成する．pk_1を検証情報であるpkと比較し，$pk_1 = pk$ならば$sk_1 = sk$である．$pk_1 \neq pk$ならば $sk_1 \neq sk$であるため 3 へ進む．
3. 端末はp_2を選び，$p_u|p_2$を擬似乱数生成器に入力する．その出力を Gen1 に入力してsk_2を得る．更に Gen2 を用い，sk_2よりpk_2を生成する．$pk_2 = pk$ならば$sk_2 = sk$である．$pk_2 \neq pk$ならば $sk_2 \neq sk$であるため 4 へ進む．
4. 端末は，検証情報を利用して，$sk_i = sk$となるまで（$p_i = p_r$となるp_iが見つかるまで）p_iの総当たりを繰り返し，skを復元する．
5. 復元されたskを用いて Dec を実行する．

ここで，記号「｜」ビット列の連結を表す．

$k_l(k)$ビットの秘密鍵$sk = sk_u|sk_r$の内，sk_uのみをユーザに渡すという方法は現実的ではないことに留意されたい．一般的に，既存の公開鍵暗号アルゴリズムの鍵長は非常に大きい（例えば RSA 暗号の鍵長は，現時点で 1024 あるいは 2048 ビット）ため，sk_uからskを復元するにあたっての計算量は非現実的な大きさとなる（例えば，skが 1024 ビット，sk_uが 64 ビットであった場合，最悪2^{960}回の総当たり試行が必要になる）．そこで，擬似乱数生成器を利用し，kビットのシード$p = p_u|p_r$の内，p_uのみをユーザに渡すという方法が採られている．

シードpのビット長（エントロピ）kをどれだけの大きさにすれば良いかを説明する．4 章で証明を行うが，計算機援用公開鍵暗号の安全性は用いる擬似乱数生成器のシードのビット長（エントロピ）kに依存する．擬似乱数生成器に入力するシードは「攻撃者に対しては秘匿されるべき秘密情報」であるため，その取扱いは共通鍵暗号における共通鍵と同等であると考えて良い．そのため，シード長kの値としては,「その時代で安全であると考えられる共通鍵暗号の鍵長」以上とすれば安全であると考えられる．

3.3 計算機援用認証コード

認証コードプリミティブに対する秘密情報補完型計算機援用セキュリティシステムの構成法を説明する．ここで，本稿における「認証コード」とは，メッセージのハッシュ値を利用して，メッセージの真正性を確認する仕組みを意味する．以降，本システムを計算機援用認証コードと呼ぶ．計算機援用認証コードは，ハッシュ関数を用いて構成される．計算機援用認証コードにおいては，メッセージのハッシュ値が「検証情報」として利用される．

メッセージをパスワードと捉えれば，認証コードはユーザ認証の仕組みと等価となる．よって，2 章で説明した計算機援用ユーザ認証は，計算機援用認証コードの 1 インスタンスである．（ただし，計算機援用ユーザ認証においては，パスワードのハッシュ値$H(p)$がメッセージの位置付けであり，そのハッシュ値$H(H(p))$が検証情報として用いられている．）

計算機援用認証コードにおいては，ハッシュ値の衝突（例えば，計算機援用ユーザ認証において，$H(H(p))$を満たすp_rは 1 つではない）に配慮し，秘密情報を復元するにあたっての総当たり試行の順序を決めておく必要がある．

計算機援用認証コードに対しては，現時点で安全性証明に至っていないため，本稿においては詳細な議論は割愛する（今後の課題である）．

4. 安全性証明

本章では，計算機援用共通鍵暗号，計算機援用公開鍵暗号の安全性証明を行う．

4.1 計算機援用共通鍵暗号

計算機援用共通鍵暗号の安全性証明として，「既知平文攻撃に対して安全な共通鍵暗号SKE」と「SKEを用いて構成した計算機援用共通鍵暗号$CASKE$」の安全性が等価であることを証明する．

定理 1

「既知平文攻撃として攻撃者がq個の平文と暗号文のペアを与えられるSKE」と「既知平文攻撃として攻撃者が$q-1$個の平文と暗号文のペアを与えられる$CASKE$」の安全性は等価である．

定理 1 の証明

「共通鍵を求めることを目的とした既知平文攻撃」を例に採り，以下の(1)と(2)を証明する．（他の目的に対する既知平文攻撃についても同様に証明できる。）

(1) q個の平文と暗号文のペアを用いて共通鍵暗号SKEの秘密鍵を無視できない確率で求める既知平文攻撃アルゴリズムAが存在するならば，$q-1$個の平文と暗号文のペアを用いて計算機援用共通鍵暗号$CASKE$の秘密鍵を無視できない確率で求める既知平文攻撃アル

ゴリズムBが存在する．

Aを利用してBを構成する（図 1）．①挑戦者Cは共通鍵skを生成し，さらに検証情報（平文と暗号文の 1 ペア(m,c)）を生成する．②Cは，検証情報(m,c)をBに入力する．③Bは既知平文攻撃を行うために，$q-1$個の平文と暗号文のペアの入手を求めるクエリーをCに送る．④Cは$q-1$個の平文と暗号文のペア$(m_1,c_1),\cdots,(m_{q-1},c_{q-1})$を生成し，⑤$B$に入力する．この時点で$B$は$A$を起動する．$B$は$A$に対する挑戦者の役割をする．⑥$A$は既知平文攻撃を行うために，$q$個の平文と暗号文のペアの入手を求めるクエリーを$B$に送る．⑦$B$は②と⑤によって得られている$q$個の平文と暗号文のペア$(m,c),(m_1,c_1),\cdots,(m_{q-1},c_{q-1})$を$A$に与える．⑧$A$は$B$に対して$sk$を出力するため，$B$はその$sk$を$C$に返答する．

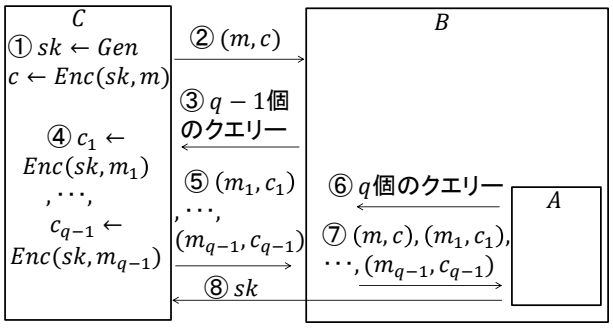

図 1　(1)の証明の概要

(2)　$q-1$個の平文と暗号文のペアを用いて計算機援用共通鍵暗号$CASKE$の秘密鍵を無視できない確率で求める既知平文攻撃アルゴリズムBが存在するならば，q個の平文と暗号文のペアを用いて共通鍵暗号SKEの秘密鍵を無視できない確率で求める既知平文攻撃アルゴリズムAが存在する．

Bを利用してAを構成する（図 2）．①挑戦者Cは共通鍵skを生成する．②Aは既知平文攻撃を行うために，q個の平文と暗号文のペアの入手を求めるクエリーをCに送る．③Cはq個の平文と暗号文のペア$(m_1,c_1),\cdots,(m_q,c_q)$を生成し，④$A$に入力する．この時点で$A$は$B$を起動する．$A$は$B$に対する挑戦者の役割をする．⑤$A$は④で得られている$q$個の平文と暗号文のペアの中の 1 ペア$(m_q,c_q)$を検証情報と見なして，$B$に与える．⑥$B$が既知平文攻撃を行うために，$q-1$個の平文と暗号文のペアの入手を求めるクエリーを$A$に送る．⑦$A$は④で得られている$q$個の平文と暗号文のペアの内，⑥の時点で$B$に与えていない$q-1$個の平文と暗号文のペア$(m_1,c_1),\cdots,(m_{q-1},c_{q-1})$を与える．⑧$B$は$A$に対して$sk$を出力するため，$A$はその$sk$を$C$に返答する．

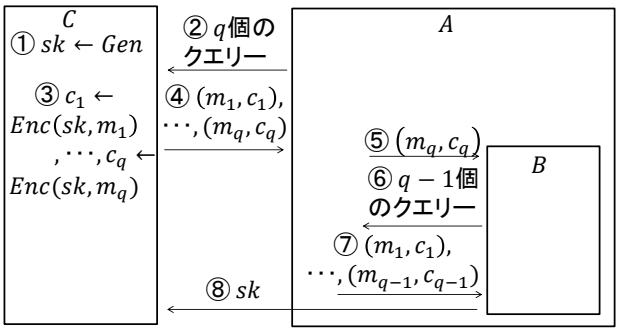

図 2　(2)の証明の概要

以上より定理 1 が証明された．

4.2　計算機援用公開鍵暗号

3.2 節で説明したように，計算機援用公開鍵暗号は疑似乱数生成器の出力から秘密鍵を生成するため，計算機援用公開鍵暗号の安全性証明は，以下の 2 段階で行う必要がある．第 1 段階が，「選択平文攻撃に対して安全な公開鍵暗号PKEの秘密鍵生成アルゴリズム Gen1」の入力として，疑似乱数（計算量的に安全な擬似乱数生成器PRZの出力）を用いた場合と真正乱数を用いた場合の安全性がほぼ同じであることの証明（定理 2）である。第 2 段階が，「選択平文攻撃に対して安全な公開鍵暗号PKE」と「PKEを用いて構成した計算機援用公開鍵暗号$CAPKE$」の安全性が等価であることの証明（定理 3）である．

定理 2

「選択平文攻撃に対して安全な公開鍵暗号PKEの秘密鍵生成アルゴリズム Gen1」に疑似乱数（計算量的に安全な疑似乱数生成器PRZの出力）を入力して秘密鍵を生成した場合の公開鍵暗号$PRGPKE$と，「選択平文攻撃に対して安全な公開鍵暗号PKEの秘密鍵生成アルゴリズム Gen1」に真正乱数を入力して秘密鍵を生成した場合の公開鍵暗号$RPKE$の安全性はほぼ等価である．

定理 2 の証明

「暗号文を識別することを目的とした選択平文攻撃（IND-CPA)」を例に採り，定理 2 を証明する．（他の目的に対する選択平文攻撃についても同様に証明できる。）

まず，攻撃者Bに対する挑戦者Cを設定し，BとCの間で実行される次のようなゲームG1（疑似乱数と真正乱数の識別ゲーム）を構成する：①Cは，ランダムに$a \in \{0,1\}$を選ぶ（0または1のいずれかが等確率で選ばれる。②$a=0$の時は，rを$\{0,1\}^{k_l}$から一様に選ばれた乱数（$k_l(k)$ビットの真正乱数）とし，Bに渡す．$a=1$の時はrを$PRG(k)$から生成される疑似乱数（$k_l(k)$ビットの疑似乱数）として，Bに渡す．③Bは，Cが①で選んだaの値（0か1か）を推測して，その答え$a^* \in \{0,1\}$をCに返答する．Bが 1/2 よりも高い確率でaの値を正答できれば（疑似乱数と真性乱数を識別できれば），Bの勝ち．

次に，「IND-CPA 安全な公開鍵暗号アルゴリズムPKE」

に対する IND-CPA 攻撃者Aを設定する．Bは，Cとのゲーム G1を行うにあたり，Aを利用することができる．Aを使役しているBをB^Aと記す．Bは，Aに対しては挑戦者となって，Aとの間でゲームG2（IND-CPA ゲーム）を実行する．

このとき，ゲームG1に対するB^Aの戦略は以下の通りとなる．

1. B^Aは，Cからrを受け取る．
2. B^Aは，rを Gen1 に入力し，その出力skを公開鍵暗号の秘密鍵とする。更に，アルゴリズム Gen2 を用いて，skに対する公開鍵pkを生成する．
3. ここで，B^AはG1からG2にスイッチする。
4. B^Aは，Aにpkを渡す．（G1からG2にスイッチ。）
5. Aは，B^Aにm_0, m_1を返す．
6. B^Aは，ランダムにチャレンジビット$b \in \{0,1\}$の値（0または1）を選び，$c^* \leftarrow Enc(pk, m_b)$を計算した上で，$A$にチャレンジ暗号文$c^*$を返す．
7. Aは，チャレンジビットbの推測値$b^* \in \{0,1\}$をB^Aに返す．
8. ここで，B^AはG2からG1に戻る。
9. B^Aは，$b \neq b^*$の時，Cに0を回答し，$b = b^*$の時，Cに1を回答する．

手順 7 において，$b = b^*$であった場合，B^Aは「Aが公開鍵暗号に対する IND-CPA 攻撃に成功した」ことが分かる．ここで用いているPKEは IND-CPA 安全な公開鍵暗号アルゴリズムであるので，真性乱数から秘密鍵skおよび公開鍵pkが生成されていた場合には，Aは IND-CPA 攻撃に成功することは不可能である．Aが IND-CPA 攻撃に成功したということは，手順 4 にてB^AからAに入力されたpkが，「擬似乱数から生成された秘密鍵sk」から生成された公開鍵であるということが演繹できる．

一方で，ゲームG1においては安全な疑似乱数生成器PRGが用いられている。安全な疑似乱数生成器の定義 1（3.2.1 節）より，いかなる攻撃者Bに対しても，

$$Adv_B^{PRG}(k) = \left|\Pr\left[B(r) \to 1 \middle| r \leftarrow \{0,1\}^{k_l(k)}\right] - \Pr[B(r) \to 1 | r \leftarrow PRG(\{0,1\}^k)]\right| < \varepsilon(k) \quad (1)$$

が成り立つ．式(1)は任意のBに対して成立するため，Aを使役してゲームG1を実行しているB^Aに対しても成り立つ．ゲームG1に対するB^Aの戦略から明らかなように，手順 9 においてB^Aが 1 を出力（r が疑似乱数であると判定）するのは，「手順 7 にて$\tilde{A}$が返してくるb^*」が$b = b^*$となる場合であるので，上式(1)は式(2)のように変形できる．

$$\max_A Adv_{B^A}^{PRG}(k) = \max_A \left|\Pr\left[b = b^* \middle| r \leftarrow \{0,1\}^{k_l(k)}\right] - \Pr[b = b^* | r \leftarrow PRG(\{0,1\}^k)]\right| < \varepsilon(k) \quad (2)$$

式(2)は，いかなる確率的多項式時間アルゴリズムAに対しても「$\Pr[b = b^* | r \leftarrow \{0,1\}^{k_l}]$と$\Pr[b = b^* | r \leftarrow PRG(s)]$の差は$\varepsilon(k)$未満である」ことを意味している．よって，いかなる A に対しても $\Pr\left[b = b^* \middle| r \leftarrow \{0,1\}^{k_l(k)}\right] + \varepsilon(k)$ は必ず $\Pr[b = b^* | r \leftarrow PRG(\{0,1\}^k)]$よりも大きいことが言え，

$$\max_A\left\{\left|\Pr[b = b^* | r \leftarrow PRG(\{0,1\}^k)]\right|\right\} < \max_A\left\{\left|\Pr\left[b = b^* \middle| r \leftarrow \{0,1\}^{k_l(k)}\right] + \varepsilon(k)\right|\right\} \quad (3)$$

が成立する．

式(3)を変形していくことにより，

$$\max_A\left\{\left|\Pr[b = b^* | r \leftarrow PRG(\{0,1\}^k)] - \frac{1}{2}\right|\right\} < \max_{\tilde{A}}\left\{\left|\Pr\left[b = b^* \middle| r \leftarrow \{0,1\}^{k_l(k)}\right] + \varepsilon(k) - \frac{1}{2}\right|\right\},$$

$$\max_A\left\{\left|\Pr[b = b^* | r \leftarrow PRG(\{0,1\}^k)] - \frac{1}{2}\right|\right\} < \max_A\left\{\left|\Pr\left[b = b^* \middle| r \leftarrow \{0,1\}^{k_l(k)}\right] - \frac{1}{2}\right|\right\} + \varepsilon(k) \quad (4)$$

が得られる．

ここで，IND-CPA 安全の定義 2（3.2.1 節）より，式(4)の左辺が公開鍵暗号$PRGPKE$（「kビットの真性乱数をシードとして生成した$k_l(k)$ビットの疑似乱数」から秘密鍵を生成した公開鍵暗号アルゴリズム）に対する IND-CPA 攻撃者Aのアドバンテージを表す式そのものであり，式(4)の右辺第 1 項が公開鍵暗号$RPKE$（$k_l(k)$ビットの真性乱数から秘密鍵を生成した公開鍵暗号アルゴリズム）に対する IND-CPA 攻撃者Aのアドバンテージの式そのものであることに気付くと，

$$Adv_{PRGPKE,A}^{IND-CPA}(k) < Adv_{RPKE,A}^{IND-CPA}(k_l) + \varepsilon(k) \quad (5)$$

であることが分かる．すなわち，両者の安全性の差はたかだか$\varepsilon(k)$である。

以上より，定理 2 が証明された．

定理 3

「選択平文攻撃に対して安全な公開鍵暗号PKE」と「PKEを用いて構成した計算機援用公開鍵暗号$CAPKE$」の安全性は等価である。

定理 3 の証明

「暗号文の識別を目的とした選択平文攻撃（IND-CPA）」を例に採り，以下の(1)と(2)を証明する．（他の目的に対する選択平文攻撃についても同様に証明できる。）

(1) 公開鍵暗号PKEに対して無視できない確率で IND-CPA 攻撃を成功させることができるアルゴリズムAが存在するならば，計算機援用公開鍵暗号$CAPKE$に対して無視できない確率で IND-CPA 攻撃を成功させることのできるアルゴリズムBが存在する．

Aを利用してBを構成する（図 3）．①IND-CPA ゲームの挑戦者Cは，公開鍵pkと秘密鍵skのペアを生成し，②攻撃者Bに公開鍵pkを入力する．③$CAPKE$においては検証情報も与える必要があるため，CはBに検証情報pkを入力する（ただし，$CAPKE$の検証情報は公開鍵と等しい）．この時点でBはAを起動する．BはAに対する挑戦者の役割をする．

④BはAにpkを入力する．⑤Aは 2 つの平文m_0, m_1をBに返す．Bはm_0, m_1をそのままCに返す．⑥Cはランダムにチャレンジビット$b \in \{0,1\}$を選び，$c^* \leftarrow Enc(pk, m_b)$を計算した上で，⑦$B$にチャレンジ暗号文$c^*$を返す．$B$は$c^*$をそのまま$A$に入力する．⑧$A$は，チャレンジビットの値（0か1か）を推測して，その答え$b^* \in \{0,1\}$をBに返答する．Bはb^*をそのままAに出力する．

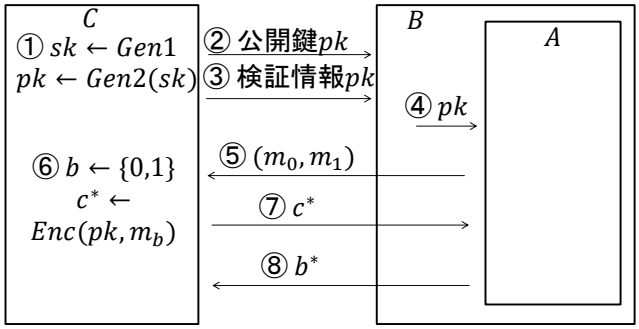

図 3　(1)の証明の概要

(2)　計算機援用公開鍵暗号$CAPKE$に対して無視できない確率で IND-CPA 攻撃を成功させることができるアルゴリズムBが存在するならば，公開鍵暗号PKEに対して無視できない確率で IND-CPA 攻撃を成功させることができるアルゴリズムAが存在する．

Bを利用してAを構成する（図 4）．①IND-CPA ゲームの挑戦者Cは，公開鍵pkと秘密鍵skのペアを生成し，②Aにpkを入力する．この時点でAはBを起動する．AはBに対する挑戦者の役割をする．③AはBに公開鍵pkを入力する．④$CAPKE$においては検証情報も与える必要があるため，AはBに検証情報pkを入力する（ただし，$CAPKE$の検証情報は公開鍵と等しい）．⑤Bは 2 つの平文m_0, m_1をAに返す．Aはm_0, m_1をそのままCに返す．⑥Cはランダムにチャレンジビット$b \in \{0,1\}$を選び，$c^* \leftarrow Enc(pk, m_b)$を計算した上で，⑦$A$にチャレンジ暗号文$c^*$を返す．$A$は$c^*$をそのまま$B$に入力する．⑧$B$は，チャレンジビットの値（0か1か）を推測して，その答え$b^* \in \{0,1\}$をAに返す．Aはb^*をそのままBに出力する．

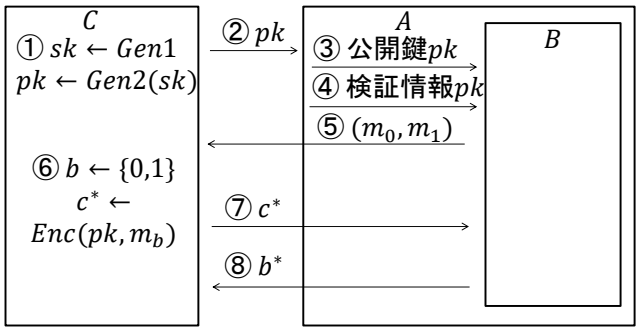

図 4　(2)の証明の概要

以上より定理 3 が証明された．

5. まとめと今後の課題

本稿では，計算機援用セキュリティシステムを「秘密情報補完」型と「秘密情報拡張」型の 2 タイプに分類した上で，共通鍵暗号，公開鍵暗号，認証コード（ユーザ認証）のプリミティブに対する秘密情報補完型計算機援用セキュリティシステムの構成法を提案し，前者 2 つの構成に対してその安全性証明を示した．

今後は，秘密情報補完型計算機援用認証コードに対する安全性証明と，秘密情報拡張型計算機援用セキュリティシステムの各種プリミティブの構成と安全性証明を行っていく予定である．

参考文献

[1] 兼子拓弥，本部栄成，高橋健太，西垣正勝．計算機援用ユーザ認証．情報処理学会論文誌. 2014, vol. 55, no. 9, p. 2072-2080.

[2] Provos, N. and Mazieres, D.. A Future-Adaptable Password Scheme. USENIX Annual Technical Conference. 1999.

[3] Turan, M. S. Barker, E. Burr, W. and Chen, L.. NIST Special Publication 800-132 Recommendation for Password-Based Key Derivation. 2010.

[4] Schneier, B.. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). Fast Software Encryption-Cambridge Security Workshop Proceedings. 1994, p. 191-204.

[5] Yao, A. C.. Theory and Applications of Trapdoor Functions(Extended Abstract).FOCS. 1982, p. 80-91.